

REMARKS

Amendments to the Specification

Pursuant to the Examiner's suggestion in the Office action dated November 5, 2003, the title of the application has been amended as follows: "METHOD AND APPARATUS FOR ALIGNING ITEMS RELATIVELY WITHIN AN ELECTRONIC DOCUMENT".

Amendments to the Claims

Claims 1-23 stand rejected. Claims 1-6, 8, 10-19 and 21-22 have been amended.

Claim 1 stands rejected under 35 U.S.C. 103 based on Microsoft Visual Basic 5.0 Programmer's Guide, Redmond, Washington, 1997 ("Microsoft") in view of U.S. Patent No. 4,974,174 issued to Kleinman ("Kleinman").

Microsoft discloses:

Displaying Print Output at a Specific Location

You can control placement of Print output by specifying the drawing coordinates, using either or both of these techniques:

- Use the Cls (clear) method to erase a form or picture box and reset the drawing coordinates to the origin (0,0).
- Set drawing coordinates with the CurrentX and CurrentY properties.

The Cls Method

All the text and graphics on the object that were created with Print and graphics methods can be deleted with the Cls method. The Cls method also resets the drawing coordinates to the origin (0,0),

which is the upper-left corner by default. For example, these statements clear:

- A picture box named Picture1:

Picture1.Cls

- The current form:

Cls

Setting Drawing Coordinates

You can set the drawing coordinates of forms and picture boxes directly with the CurrentX and CurrentY properties. For example, these statements reset the drawing coordinates to the upper-left corner for Picture1 and for the current form:

- A picture box named Picture1:

Picture1.CurrentX = 0

Picture1.CurrentY = 0

- The current form:

CurrentX = 0

CurrentY = 0

Any new text you print appears on top of any text and graphics already at that location. To erase text selectively, draw a box with the Line method and fill it with the background color. Keep in mind that the drawing coordinates specified by CurrentX and CurrentY usually change location when you use a graphics method.

By default, forms and picture boxes use a coordinate system where each unit corresponds to a twip (1,440 twips equal an inch, and approximately 567 twips equal a centimeter). You may want to change the ScaleMode property of the form, picture box, or Printer object from twips to points, because text height is measured in points. Using the same unit of measure for the text and for the object where you will print the text makes it easier to calculate the position of the text.

(Pages 547-548). Microsoft also discloses:

Understanding the Coordinate System

Every graphical operation described in this chapter (including resizing, moving, and drawing) uses the coordinate system of the drawing area or container. Although you can use the coordinate system to achieve graphical effects, it is also important to know how to use the coordinate system to define the location of forms and controls in your application.

The coordinate system is a two-dimensional grid that defines locations on the screen, in a form, or other container (such as a picture box or Printer object). You define locations on this grid using coordinates in the form:

(x,y)

The value of x is the location of the point along the x-axis, with the default location of 0 at the extreme left. The value of y is the location of the point along the y-axis, with the default location of 0 at the extreme top. This coordinate system is illustrated in Figure 12.3.

The following rules apply to the Visual Basic coordinate system:

- When you move or resize a control, you use the coordinate system of the control's container. If you draw the object directly on the form, the form is the container. If you draw the control inside a frame or picture box, the frame or the control is the container.
- All graphics and Print methods use the coordinate system of the container. For example, statements that draw inside a picture box use the coordinate system of that control.
- Statements that resize or move a form always express the form's position and size in twips.

When you create code to resize or move a form, you should first check the Height and Width properties of the Screen object to make sure the form will fit on the screen.

- The upper-left corner of the screen is always (0,0). The default coordinate system for any container starts with the

(0,0) coordinate in the upper-left corner of the container.

The units of measure used to define locations along these axes are collectively called the *scale*. In Visual Basic, each axis in the coordinate system can have its own scale.

You can change the direction of the axis, the starting point, and the scale of the coordinate system, but use the default system for now. “Changing an Object’s Coordinate System” later in this chapter discusses how to make these changes.

(Pages 557-558). Microsoft therefore discloses a conventional coordinate system, where each position of an object is specified by a “ coordinate system [that] is a two-dimensional grid that defines locations on the screen. . . . You define locations on this grid using coordinates in the form: (x,y). . . . The upper-left corner of the screen is always (0,0).”

Microsoft does not disclose “for each further item . . . positioning the item within the electronic document in accordance with the relative tab stop associated with the further item and the position of each previously positioned item associated with the same relative tab stop as the further item,” as recited in claim 1.

Kleinman discloses:

The present invention includes a method of displaying multiple objects on a display terminal wherein each of the objects is positioned by reference to a previous object.

(Column 2, lines 61-64). Kleinman also discloses:

The same alignment designations are used, and the same relative positioning results, although the absolute position of the different selections on the display may change as a result of the different text lengths. Thus, for example, changing the menu system shown in FIG. 6 from English to German, which might require the use of words having many more letters, may be done by merely changing the description text.

(Column 6, lines 60-67). Kleinman does not disclose “for each further item ... positioning the item within the electronic document in accordance with the relative tab stop associated with the further item and the position of each previously positioned item associated with the same relative tab stop as the further item,” as recited in claim 1.

Even if Microsoft and Kleinman were combined, the combination would neither teach nor suggest “for each further item ... positioning the item within the electronic document in accordance with the relative tab stop associated with the further item and the position of each previously positioned item associated with the same relative tab stop as the further item,” as recited in claim 1.

Therefore, applicants submit that claim 1 is patentable over Microsoft in view of Kleinman. Given that claims 2-10 depend from claim 1, applicants submit that these claims are also patentable over Microsoft in view of Kleinman.

Claim 11 stands rejected under 35 U.S.C. 103 based on Microsoft in view of Kleinman. Microsoft and Kleinman, alone or in combination, neither disclose nor suggest an apparatus that can “position the item within the electronic document in accordance with the relative tab stop and the position of each previously positioned item having the same relative tab stop,” as recited in claim 11. Therefore, applicants submit that claim 11 is patentable over Microsoft in view of Kleinman. Given that claims 12-13 depend from claim 11, applicants submit that these claims are also patentable over Microsoft in view of Kleinman.

Claim 14 stands rejected under 35 U.S.C. 103 based on Microsoft in view of Kleinman. Microsoft and Kleinman, alone or in combination, neither disclose nor suggest “positioning the

item within the electronic document in accordance with the relative tab stop and the position of each previously positioned item having the same relative tab stop,” as recited in claim 14.

Therefore, applicants submit that claim 14 is patentable over Microsoft in view of Kleinman.

Given that claims 15-23 depend from claim 14, applicants submit that these claims are also patentable over Microsoft in view of Kleinman.

CONCLUSION

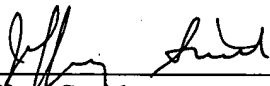
On the basis of the above remarks, reconsideration and allowance of the claims is believed to be warranted and such action is respectfully requested. If the Examiner has any questions or comments, the Examiner is respectfully requested to contact the undersigned at the number listed below.

If the Commissioner determines that additional fees are due or that an excess fee has been paid, the Patent Office is authorized to debit or credit (respectively) Deposit Account No. 50-2518, billing reference no. 7011102001.

Respectfully submitted,

Bingham McCutchen LLP

Dated: May²⁶, 2004

By: 
Jeffrey Smith
Reg. No. 39,377

Three Embarcadero Center, Suite 1800
San Francisco, CA 94111-4067
Telephone: (650) 849-4422
Telefax: (650) 849-4800